Orientation-aware Deep Neural Network for Real Image Super-Resolution

Chen Du*, He Zewei*, Sun Anshun, Yang Jiangxin, Cao Yanlong, Cao Yanpeng†
School of Mechanical Engineering
Zhejiang University
{dudaway,zeweihe,as_sun,yangjx,sdcaoyl,caoyp}@zju.edu.cn

Tang Siliang
School of Computer Science and Technology
Zhejiang University
siliang@zju.edu.cn

Michael Ying Yang
SUG - Scene Understanding Group
University of Twente
michael.yang@utwente.nl

Abstract

Recently, Convolutional Neural Network (CNN) based approaches have achieved impressive single image super-resolution (SISR) performance in terms of accuracy and visual effects. It is noted that most SISR methods assume that the low-resolution (LR) images are obtained through bicubic interpolation down-sampling, thus their performance on real-world LR images is limited. In this paper, we proposed a novel orientation-aware deep neural network (OA-DNN) model, which incorporate a number of orientation feature extraction and channel attention modules (OAMs), to achieve good SR performance on real-world LR images captured by a single-lens reflex (DSLR) camera. Orientation-aware features extracted in different directions are adaptively combined through a channel-wise attention mechanism to generate more distinctive features for high-fidelity recovery of image details. Moreover, we reshape the input image into smaller spatial size but deeper depth via an inverse pixel-shuffle operation to accelerate the training/testing speed without sacrificing restoration accuracy. Extensive experimental results indicate that our OA-DNN model achieves a good balance between accuracy and speed. The extended OA-DNN∗+ model further increases PSNR index by 0.18 dB compared with our previously submitted version. Codes will be made public after publication.

1. Introduction

Single image super-Resolution (SISR) aims to recover corresponding high-resolution (HR) image from a single low-resolution (LR) image. SISR has attracted considerable attention from both the academic and industrial communities in recent years, resulting in extensive applications such as security surveillance, autonomous driving, and medical analysis. Recently, Convolutional Neural Network (CNN) based SISR methods have achieved impressive performance by learning the mapping between low-frequency signals (object semantics) and high-frequency signals (object details) from substantial pairs of HR and LR training images.

In most CNN-based SISR approaches [14, 20, 21, 36], the LR training images are typically generated by down-sampling the HR ones via bicubic interpolation. Their performance on real-world captured LR images is not satisfactory. The challenge is two-folder. First, the down-sampling process of HR images remains unknown and device-dependent. Moreover, undesired artifacts (e.g., sensor noise, motion blur, and pixel shifts) are typically presented in real-world captured LR images. Second, the captured LR images sometimes are automatically up-sampled by the image acquisition device (e.g., DSLR camera)[1]. Directly applying previous deep CNN models (e.g., EDSR [24], RDN [44] and RCAN [43]) on the up-scaled LR images demands Graphics processing units (GPUs) with extremely large memories.

To tackle the problems mentioned above, we proposed a novel orientation-aware deep neural network (OA-DNN) model to super-resolve real-world captured LR images. The proposed OA-DNN model contains a number of orientation-aware feature extraction and channel attention modules (OAMs), in which three well-designed convolutional layers (i.e., 5×1 horizontal conv. layer, 1×5 vertical conv. layer and 3×3 diagonal conv. layer) are deployed to extract orientation-aware features in different directions. OAM also contains a channel attention mech-

*Authors contributed equally
†Corresponding author

[1] In the NTIRE 2019 Real Super-Resolution challenge dataset, the captured LR and HR images have the same resolution.

anism, which is initially proposed by Hu et al. [12], to compute channel-wise weights for adaptive fusion of the extracted orientation-aware features, generating more distinctive feature maps for high-fidelity recovery of image details. To efficiently process up-sampled LR image in the NTIRE 2019 Real Super-Resolution challenge dataset, we reshape the input image via an inverse pixel-shuffle operation (de-pixel-shuffle [27, 37]) into smaller spatial size but deeper depth. Spatial features are rearranged into multiple channels to accelerate the training/testing speed and alleviate the burden on GPU memory, while image pixel values are well preserved for inferences in the following convolutional layers. Such operation significantly reduces the memory requirement of GPUs, speeds up the training/testing processes and surprisingly boosts the SR accuracy. The main contributions of this paper are as follows.

- We present a novel feature extraction technique using three well-designed convolutional layers (5×1 horizontal conv., 1×5 vertical conv., and 3×3 diagonal conv.) to extract orientation-aware features in different directions. This is the first work that employs directional features for super-resolution task.

- A channel attention mechanism is utilized to adaptively fuse the extracted orientation-aware features, generating more distinctive feature maps for accurate SISR of real-world LR images. Different from previous methods [43, 16], we place the channel attention before ReLU to allow more information pass through activation for better performance.

- The de-pixel-shuffle operation, which was previously used for object detection, is successfully adopted for the SISR task and lead to higher SR accuracy and faster execution speed.

The remainder of this paper is organized as follows. We first review a number of CNN-based SISR methods in Sec. 2. Then Sec. 3 provides details and important components of our OA-DNN. Qualitative and quantitative comparisons are conducted in Sec. 4 to show the effectiveness of our OA-DNN and Sec. 5 concludes this paper.

2. Related Work

Single image super-resolution (SISR) refers to the task of recovering corresponding HR image from only one LR observation of the same scene. Over the past decades, substantial approaches [9, 2, 23, 30, 1, 40, 40, 31, 34, 35, 13] have been proposed to solve this problem.

Currently, deep-learning-based/CNN-based methods [6, 8, 17, 18, 32, 33, 36, 44, 11, 4] have demonstrated remarkable results by learning the LR-to-HR mapping function via numerous representative example pairs. In this paper, we focus on CNN-based SISR methods.

Dong et al.[6, 7] proposed the super-resolution convolutional neural network (SRCNN), which is the first CNN-based method with a light-weight structure (three layers). By following this pioneering work, Kim et al. [17] extended SRCNN to 20 layers and employed residual learning/adjustable clip gradients to ease the training process. The same authors also proposed DRCN [18] which establishes recursive units to share parameters and utilizes skip-connections to ease the difficulty of training the model. Lai et al. [20] proposed LapSRN to progressively reconstruct the sub-band residuals of high-resolution images and generate multi-scale predictions just through one feed-forward pass, thereby facilitated resource-aware applications.

To achieve faster speed, FSRCNN [8] introduced the deconvolution layer into SRCNN model, so the mapping function is learned directly from the original low-resolution image (without interpolation) to the high-resolution one. ESPCN [29] introduced an efficient sub-pixel convolution layer which can learn an array of up-scaling filters to upscale the final LR feature maps into the HR output. These two methods upscale the resolution at the end of models, therefore time-consuming operations are performed on LR spaces.

To achieve higher reconstruction accuracy, recent methods further increase the depth or utilize complicated architecture. DRRN [32] proposed a very deep CNN model and adopted residual learning both in global and local manners to mitigate the difficulty of training. In MemNet, Tai et al. [33] introduced a memory block which could control how much of the previous states should be reserved, and decide how much of the current state should be stored. SR-DenseNet [36] introduced dense skip connections into CNN model so the feature maps of each layer are propagated into all subsequent layers, providing an effective way to combine the low-level features and high-level features to boost the reconstruction performance. In DBPN, Muhammad et al. [10] constructed mutually connected up- and down-sampling stages, each of which represents different types of image degradation and high resolution components. WDSR [41] utilized a slim identity mapping pathway with wider channels before activation in each residual block and led to a better accuracy. Wang et al. [38] established DBDN which extended previous intra-block dense connection approaches by including novel inter-block dense connections. MSRN [22] adopted a multi-scale residual structure to fully extract the features and introduced different size of convolution kernels to adaptively detect the image features in different scales and then interact with each other to get the most efficacious image information. TSCN [14] proposed a two-stage convolutional network to estimate the desired high-resolution image from the corresponding low-resolution im-

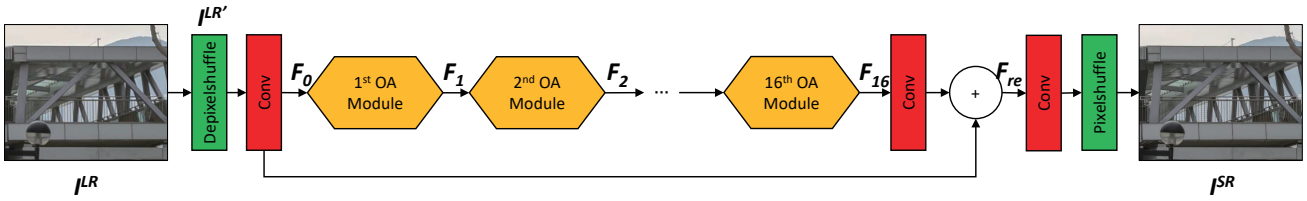

Figure 1. The architecture of our proposed OA-DNN. Let I^{LR} denote the LR input, the pixels of I^{LR} are rearranged by the de-pixel-shuffle operator into $I^{LR'}$, which has smaller size but deeper channels. 16 OAMs are used to extract directional features for inferring LR-to-HR mapping function. Then, global residual learning is added to ease the training process. Finally, we use one convolution layer and the pixel-shuffle operation to reconstruct final output I^{SR}.

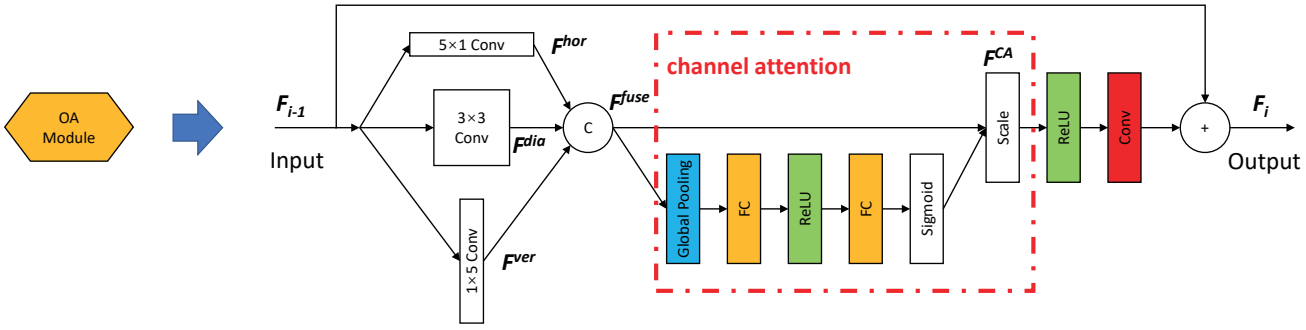

Figure 2. The architecture of our backbone OAM. The input F_{i-1} is firstly fed to three different directional convolutional layers. Then the extracted orientation-aware features F^{hor}, F^{ver} and F^{dia} are fused and then passed through a channel attention unit [12] which could adaptively compute the channel-wise weights and assign the weights to corresponding channels. At last, local residual learning is also added to ease the training process.

age.

More recently, channel attention [12] are utilized in super-resolution methods. SESR [16] introduced channel attention unit into their networks to model the interdependencies and relationships between channels. RCAN [43] proposed a residual in residual structure and introduced the channel attention mechanism to adaptively rescale channel-wise features by considering interdependencies among channels. It is noted that although deeper and more complex networks could achieve state-of-the-art reconstruction results, they will also lead to computational complexity and cost lots of time during training or testing.

Different from previous methods which aim to recover accurate PSNR, some novel works contribute to obtain photo-realistic reconstructions. SRGAN [21] introduced a perceptual loss function which consists of an adversarial loss and a content loss into their network to reconstruct photo-realistic results. In EnhanceNet, Mehdi et al. [28] proposed a novel application of automated texture synthesis in combination with a perceptual loss focusing on creating realistic textures rather than optimizing for a pixel accurate reproduction of ground truth images during training and achieved good reconstructions. However, the generated high-frequency details may be fake texture patterns, which are not suitable for some applications demanding accurate information.

3. Approach

Fig. 1 shows the workflow of our proposed OA-DNN. We firstly provide details of OAM which is used as the backbone in the proposed OA-DNN model. The overall architecture of OA-DNN are then presented and some techniques used to improve the performance of OA-DNN are also discussed.

3.1. Orientation-aware Feature Extraction and Channel Attention Module

Fig. 2 illustrates the basic module OAM in our OA-DNN. In most CNN based SISR algorithms, 3×3 convolutional kernels are utilized for feature extraction (e.g., VDSR [17], DRCN [18], DRRN [32], SRDenseNet [36], EDSR [24], TSCN [14]). Our OAM creatively embeds a 3×3 convolutional layer and two 1-D convolutional layers (i.e., 5×1 and 1×5), which are seldomly used in the SISR task, to extract features in the diagonal, horizontal and vertical directions, respectively. let F_{i-1} denotes the

input features of a single OAM, the orientation-aware features $\{F^{hor}, F^{ver}, F^{dia}\}$ are computed as:

$$F^{hor} = Conv_{5\times1}(F_{i-1}), \tag{1}$$

$$F^{ver} = Conv_{1\times5}(F_{i-1}), \tag{2}$$

$$F^{dia} = Conv_{3\times3}(F_{i-1}), \tag{3}$$

where $Conv_{5\times1}$, $Conv_{1\times5}$ and $Conv_{3\times3}$ represent the convolutional layers with horizontal kernel, vertical kernel and diagonal kernel, respectively. With this design, our OAM is capable of extracting features in different directions. Then, the extracted orientation-aware features are fused through a channel-wise concatenation operation as:

$$F^{fuse} = [F^{hor}, F^{ver}, F^{dia}], \tag{4}$$

where F^{fuse} is the fused orientation-aware feature and the $[\cdot]$ indicates the channel-wise concatenation operation.

Channel attention [12] provides an effective technique to recalibrate channel-wise features adaptively by explicitly modeling interdependencies between channels. Previous studies have proven the effectiveness of channel attention block [25, 5, 43, 16] in the task of super-resolution. In the proposed OAM, we adopt the channel attention mechanism described in [16] to adaptively combine orientation-aware features to generate more distinctive features as:

$$F^{CA} = CA(F^{fuse}) * F^{fuse} \tag{5}$$

where F^{CA} denotes the enhanced features using the channel attention mechanism and $CA(\cdot)$ are the calculated channel-wise weights. The computed F^{CA} is then activated by a rectified linear unit (ReLU) and fed to another 3×3 convolutional layer. In addition, residual learning is also deployed to ease the training process. The output F_i of the i_{th} OAM block is computed as:

$$F_i = F_{i-1} + Conv_{3\times3}(max(0, F^{CA})), \tag{6}$$

where $max(\cdot)$ indicates the ReLU activation operation. As pointed out in [41], the features F^{CA} before ReLU activation is wider than subsequent convolutional layers, which allows more information pass through ReLU while still keeps highly non-linearity of CNN. The effectiveness of the proposed orientation-aware feature extraction and channel attention based fusion are systematically evaluated in Sec. 4.3.

3.2. De-pixel-shuffle

Real-world captured LR images sometimes are automatically up-sampled by the image acquisition device (e.g., In the NTIRE 2019 Real Super-Resolution challenge dataset, LR and HR images captured by a DSLR camera have the same resolution). Consequently, directly applying previous state-of-the-art methods (e.g., EDSR [24], RDN [44] and RCAN [43]) demands GPUs with extremely large memories [2]. A feasible way for solving this problem is to add a down-sampling process to reduce the spatial size of input image. The subsequent convolutional operations can be conducted on LR space, which will largely save the GPU memory and running time. However, early stage down-sampling operation will lose important image information and lead to poor performance of SR.

Shi et al. [29] introduced an efficient pixel-shuffle operation, which upscales the spatial size via the rearrangement of the features in multiple channels. An inverse operation (de-pixel-shuffle) can be further used to reduce the spatial size of feature maps at the cost of adding multiple channels. Therefore, image information is well preserved for inferences in the following convolutional layers. As illustrated in Fig. 3, de-pixel-shuffle rearranges the input features of size $H \times W \times C$ into size $\frac{H}{r} \times \frac{W}{r} \times r^2C$ (r denotes the scaling-factor). In our implementation, we set r to 2 and the evaluation experiments are provided in Sec. 4.3.

3.3. Basic Network Architecture

As illustrated in Fig. 1, our OA-DNN aims to learn the end-to-end mapping function f from LR input I^{LR} to HR ground truth I^{GT}, which consists of 16 OAMs, three convolutional layers, a global residual learning, a de-pixel-shuffle operation, and a pixel-shuffle operation. Given a LR input I^{LR}, an inverse pixel-shuffle operation is firstly utilized to systematically rearrange the pixels into channels to reduce the spatial size. Specifically, an input of size $W \times H \times C$ is converted to $\frac{H}{r} \times \frac{W}{r} \times r^2C$. See more details in Sec. 3.2. The formulation of de-pixel-shuffle can be expressed as

$$I^{LR'} = DPS(I^{LR}), \tag{7}$$

where $DPS(\cdot)$ denotes the de-pixel-shuffle operation and $I^{LR'}$ denotes the shuffled LR image vector. Then, a 3×3 convolutional layer is embedded to extract high-dimensional features from $I^{LR'}$ as:

$$F_0 = Conv_{3\times3}(I^{LR'}), \tag{8}$$

where F_0 denotes the extracted high-dimensional feature vectors. After feature extraction, F_0 is fed into stacked OAMs and output of the i_{th} OAM F_i can be expressed as:

$$F_i = OAM_i(F_{i-1}), i \in \{1, 2, \cdots, 16\}, \tag{9}$$

where $OAM(\cdot)$ denotes the operations of a single OAM. We also employ the global residual learning to ease the training process. Before that a 3×3 convolutional layer is embedded. The formulation is as follows:

$$F_{re} = F_0 + Conv_{3\times3}(F_{16}), \tag{10}$$

[2]The upsampling parts in these methods should be removed for applying on NTIRE2019 Real Super-Resolution Challenge.

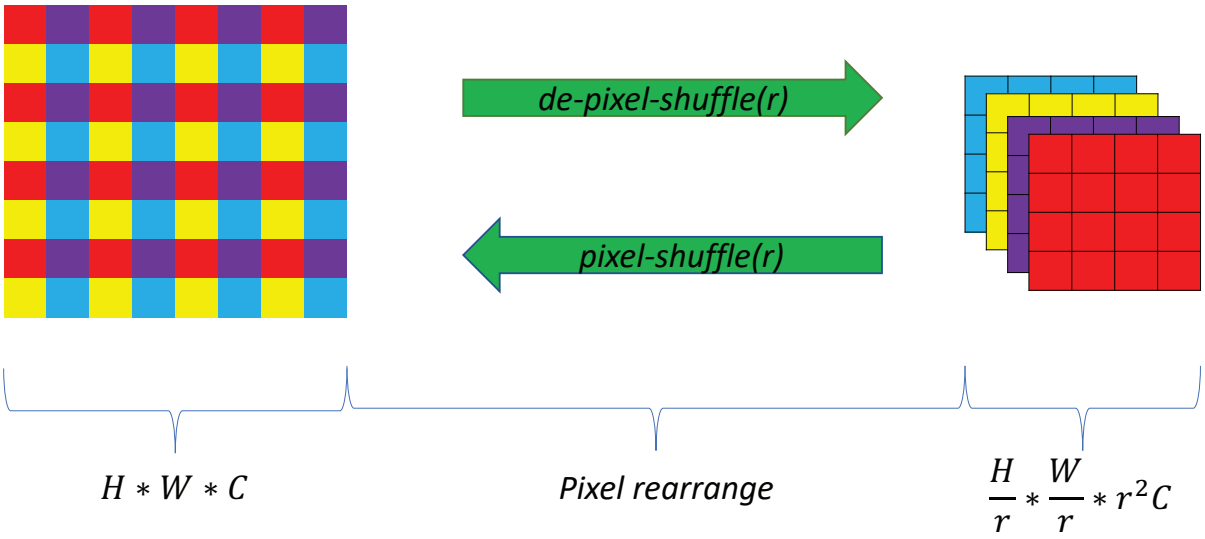

Figure 3. The principles of pixel-shuffle and de-pixel-shuffle. The pixels of a feature map can be rearranged into larger spatial size but fewer channels through the pixel-shuffle operation [29]. On the contrary, pixels of an image can also be rearranged into smaller spatial size but deeper channels through the de-pixel-shuffle operation. r denotes the scale factor.

where F_{re} denotes the high-dimensional vector for reconstructing the super-resolved HR image I^{SR}. In the reconstruction phase, the high-dimensional vector F_{re} is channel-wisely shrunk to the size of $\frac{H}{r} \times \frac{W}{r} \times r^2C$ before the pixel-shuffle operation $PS(\cdot)$. The shrink can be realized via a 3×3 convolutional layer by setting the output channel to the desired value. Finally, pixel-shuffle rearranges the pixels to form the final super-resolved image I^{SR} as:

$$I^{SR} = PS(Conv_{3\times3}(F_{re})). \tag{11}$$

3.4. Deep Supervision

Our basic OA-DNN architecture contains 16 OAMs, which consist of many convolutional layers and will unfavorably cause the gradient vanishing problem. To solve this problem and further enhance feature maps extracted in different layers, we send the output of selected OAMs (i.e., 4_{th}, 8_{th}, 12_{th} and 16_{th} in our implementation) to the reconstruction part during training phase to generate 4 predictions, as shown in Fig. 4. Different from the deep supervision strategy adopted in [18, 33], the generated predictions are not further fused to form the final output. We only make use of the prediction based on features of 16_{th} OAM as our final SISR output. It is worth mentioning that this deep supervision strategy only takes a little more time during training but cost no extra time/computational increase during the testing phase.

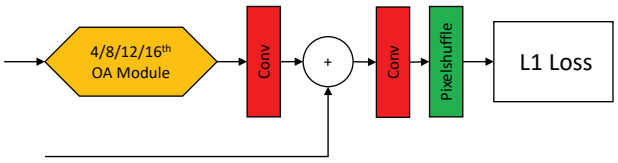

Figure 4. Deep supervision: we add supervisions after 4_{th}, 8_{th}, 12_{th} and 16_{th} OAMs.

3.5. Loss Function

Loss function computes the pixel-wise difference between the super-resolved image I^{SR} and the ground truth I^{GT}, which drives the back-propagation to update the weights and biases of CNN. Most deep learning based SR methods [6, 17, 18, 32, 33] adopt L_2 (i.e., mean square error loss or Euclidean loss) as the training loss. The main reason behind its popularity is that the calculation of L_2 loss is similar with a major SR evaluation indicator - PSNR. The loss function $\mathcal{L}_{L_2}$ is defined as:

$$\mathcal{L}_{L_2}(P) = \sum_{p\in P} ||I^{SR}(p) - I^{GT}(p)||_2^2, \tag{12}$$

where $||\cdot||_2$ denotes the L_2 norm. Nevertheless, Lim et al. [24] experimentally reported that L_1 is a better option than $L2$. Similar with $\mathcal{L}_{L_2}$, the loss function $\mathcal{L}_{L_1}$ is defined as:

$$\mathcal{L}_{L_1}(P) = \sum_{p\in P} ||I^{SR}(p) - I^{GT}(p)||_1, \tag{13}$$

where $||\cdot||_1$ denotes the L_1 norm. In our method, we choose the L_1 loss which provides a large back-propagated derivative to speed up the training process at the beginning. With the training going on, most of the residual values approach zeros and we use L_2 loss with smaller back-propagated derivatives for the fine solution searching.

3.6. Geometric Self-ensemble

In the testing phase, following EDSR [24, 35], the self-ensemble strategy is adopted to further improve the SR performance. Specifically, when testing, the input image is rotated to generate three other augmented inputs. After achieving corresponding super-resolved images, the inverse transform is applied to get the original geometry. Finally, we average the transformed outputs to obtain the final result. Compared with previous methods [24, 35] which generate seven augmented inputs via rotation and horizontal flipping, our method only uses three augmented inputs and experimentally achieves similar performance using less running time.

4. Experiments

4.1. Dataset and Metrics

Dataset: For NTIRE2019 Real Super-Resolution Challenge, the organizers published a novel dataset of real low and high resolution paired images, which are obtained in diverse indoor and outdoor environments by DSLR cameras. The dataset consists of 100 pairs of LR images and their corresponding ground truth HR ones. These pairs are divided into 60 pairs for training, 20 pairs for validation and another 20 pairs for testing. Each image has a pixel resolution no smaller than 1000×1000. As the test dataset ground truth is not released, we report the performances and compare with state-of-the-art methods on the validation dataset. To expand our training dataset, two data augmentation techniques are utilized including (1) Rotation: rotate image by $90°$, $180°$, or $270°$. (2) Flipping: flip images horizontally. After data augmentation, we randomly crop these images into 48×48 patches for training our OA-DNN.

Metrics: Peak signal-to-noise-ratio (PSNR) and structural similarity index (SSIM) [39] are used for SR performance evaluation. Both metrics are calculated on RGB channels without crop pixels near image boundary according to the scoring scripts provided by the NTIRE 2019 Real Super-Resolution Challenge organizers.

4.2. Implementation Details

We implement our OA-DNN with Caffe[15] platform and train this model by optimizing L_1 loss function on a single NVIDIA Quadro P6000 GPU with Cuda 9.0 and Cudnn 7.1 for 20 epochs. When training our model, we only consider the luminance channel (Y channel of YCbCr color space) in our experiments. Adam[19] solver is utilized to optimize the weights by setting $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\varepsilon = 10^{-8}$. In each training batch, we randomly sample 64 patches with size of $48 \times 48 \times 1$. By employing the de-pixel-shuffle operation, discussed in section 3.2, the patches are reshaped to $24 \times 24 \times 4$. The initial learning rate is set to 10^{-4} and halved after 15 epochs. After 20 epochs, we fine-tune our model for one more epoch by optimizing L_2 loss function. Training our final OA-DNN for real image super-resolution approximately takes two days.

4.3. Model Analysis

As illustrated in Tab. 1, we set up the following ablation experiments to explore the advantages of our proposed OAM and de-pixel-shuffle operation. Experiment A (Exp-A): We utilize the residual module from EDSR [24] to replace our OAM, and remove the de-pixel-shuffle operation. Experiment B (Exp-B): On the basis of Exp-A, the de-pixel-shuffle operation is added. Experiment C (Exp-C): On the basis of Exp-A, OAM is adopted as the backbone. Experiment D (Exp-D): Take OAM as the backbone and add the de-pixel-shuffle operation. All the experiments are performed on our basic network architecture.

Compare Exp-B with Exp-A, we surprisingly observed that the PSNR value increases from 29.11 dB to 29.18 dB by utilizing de-pixel-shuffle operation. Another benefit is the computational cost will be largely reduced by taking convolutional operations on small spatial size. The running time decreases from 1.2844s to 0.5352s. Compare Exp-C with Exp-A, the PSNR value increases from 29.11 dB to 29.28 dB by adopting OAM instead of the residual module from [24]. The proposed OAM can extract directional features and fuse them for learning better mapping. More parameters and complicated structure (i.e., channel attention mechanism) will unavoidably consume extra running time (1.2844s $\rightarrow$ 4.0578s). By adding de-pixel-shuffle operation to Exp-C, the PSNR value reaches 29.35 dB (0.24 dB higher than Exp-A, which is a significant improvement in SISR). Meanwhile, the running time only increases from 1.2844s to 1.6636s.

Based on Exp-D, we also explore the effectiveness of our tricks: (1) deep supervision, (2) Fine-tune with L_2 loss, and (3) geometric self-ensemble. Tab. 2 shows the quantitative results of adding different tricks. Obviously, all the three tricks can boost the performance (PSNR: 29.35 dB $\rightarrow$ 29.42 dB $\rightarrow$ 29.47 dB $\rightarrow$ 29.59 dB; SSIM: 0.8599 $\rightarrow$ 0.8614 $\rightarrow$ 0.8628 $\rightarrow$ 0.8652.). It is noted that deep supervision and fine-tune with L_2 loss improve performance without triggering any computational cost during testing.

4.4. Comparisons with State-of-the-arts

To prove the effectiveness of our proposed OA-DNN, two CNN-based methods (VDSR [17] and DRRN [32])

Figure 5. Qualitative comparisons of image "cam1-05" from validation dataset provided by the NTIRE 2019 organizers. We re-train VDSR and DRRN on this real SR dataset to obtain their results. Please zoom in on screen for better visualization.

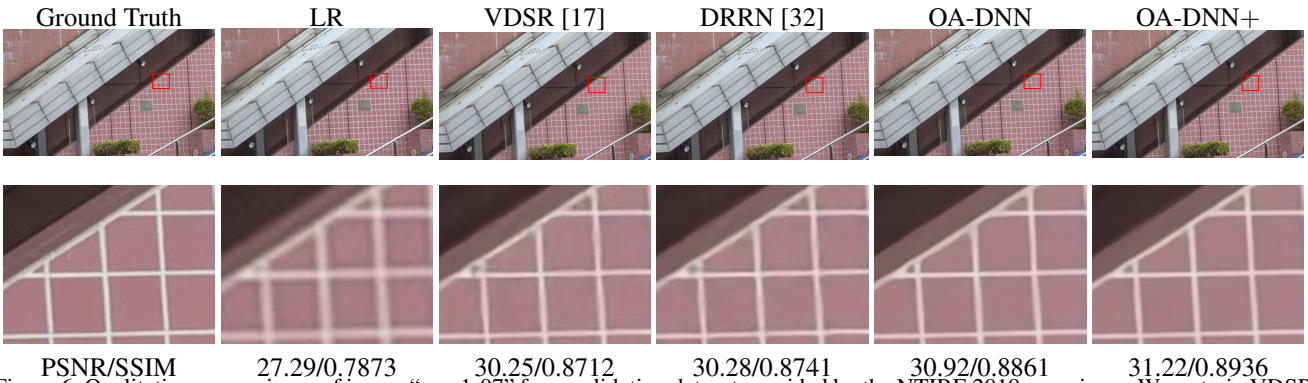

Figure 6. Qualitative comparisons of image "cam1-07" from validation dataset provided by the NTIRE 2019 organizers. We re-train VDSR and DRRN on this real SR dataset to obtain their results. Please zoom in on screen for better visualization.

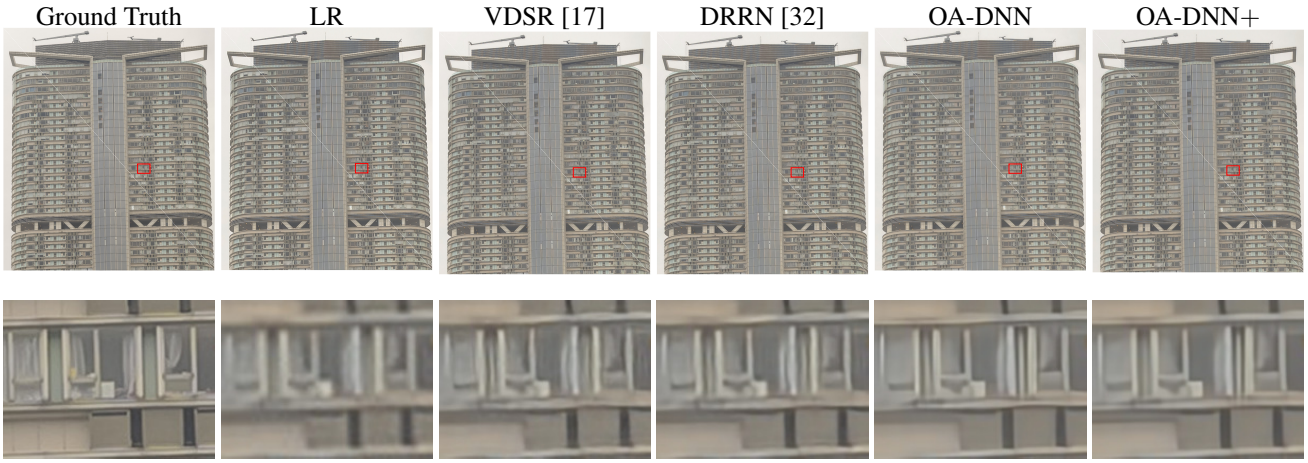

Figure 7. Qualitative comparisons of image "cam2-05" from validation dataset provided by the NTIRE 2019 organizers. We re-train VDSR and DRRN on this real SR dataset to obtain their results. Please zoom in on screen for better visualization.

are retrained using the real SR dataset provided by the NTIRE 2019 organizers. The quantitative results are shown in Tab. 3 and the qualitative comparisons are illustrated in Fig. 5, Fig. 6 and Fig. 7.

From Tab. 3, we can get the conclusion that our OA-DNN achieves the best performance among state-of-the-art SISR methods. In addition, Fig. 5, Fig. 6 and Fig. 7 indicate that our proposed OA-DNN recovers relatively sharper edges, while others only produce blurry results. By employing directional features from different orientations, OA-DNN can better reconstruct the line pattern.

Table 1. The quantitative SR results on validation dataset with different combinations of OAM and de-pixel-shuffle. The PSNR and SSIM values are calculated according to the scoring scripts provided by the NTIRE 2019 organizers.

	Different Combinations			
	Exp-A	Exp-B	Exp-C	Exp-D
OAM	×	×	✓	✓
De-pixel-shuffle	×	✓	×	✓
PSNR(dB)	29.11	29.18	29.27	**29.35**
SSIM	0.8550	0.8560	0.8571	**0.8599**
Time (s)	1.2844	0.5352	4.0578	1.6636

Table 2. The quantitative SR results on validation dataset with different tricks. The PSNR and SSIM values are calculated according to the scoring scripts provided by the NTIRE 2019 organizers.

Different tricks	Settings			
Baseline	✓	✓	✓	✓
Deep Supervision	×	✓	✓	✓
L_2 Fine-tune	×	×	✓	✓
Self-ensemble	×	×	×	✓
PSNR(dB)	29.35	29.42	29.47	**29.59**
SSIM	0.8599	0.8614	0.8628	**0.8652**
Time (s)	1.6636	1.6636	1.6636	6.1097

Table 3. The quantitative results on validation dataset with VDSR [17] and DRRN [32]. The PSNR and SSIM values are calculated according to the scoring scripts provided by the NTIRE 2019 organizers.

Different Methods	PSNR (dB)	SSIM
VDSR [17]	29.10	0.8524
DRRN [32]	29.13	0.8538
OA-DNN	29.47	0.8628
OA-DNN+	**29.59**	**0.8652**

4.5. Enhanced Performance of our OA-DNN

After the NTIRE 2019 Real Super-Resolution Challenge submission deadline, we further modified the training settings of our submitted OA-DNN to improve the performance further. Three simple modifications are performed:

- We re-train our OA-DNN with RGB input patches and pre-process all the training patches by subtracting the mean RGB value of the training dataset.
- Larger patch size (128×128) are adopted to learn the end-to-end mapping function.
- More modules (20 OAMs) are utilized to constitute our OA-DNN.

We denote the model using new training setting as OA-DNN$*$, which has improved performance than the submitted version of our OA-DNN+. Tab. 4 shows the comparative results of OA-DNN, OA-DNN+, OA-DNN$*$ and OA-DNN$*$+. **It's worth mentioning that PSNR value of our OA-DNN$*$ reaches 29.63 dB with faster testing speed than our submitted OA-DNN+. Our ultimate OA-DNN$*$+ even achieves PSNR 0.18 dB improvement than our submitted version.** Our extended OA-DNN$*$ has already achieved higher PSNR values (29.63 dB $>$ 29.59 dB) than our submitted OA-DNN+ with about $\frac{1}{3}$ running time (2.0251s $<$ 6.1097s).

Table 4. Comparative results of our OA-DNN, OA-DNN+, OA-DNN$*$ and OA-DNN$*$+.

Methods	PSNR (dB)	Running time (s)
OA-DNN	29.47	**1.6636**
OA-DNN+	29.59	6.1097
OA-DNN$*$	29.63	2.0251
OA-DNN$*$+	**29.77**	8.1023

5. Conclusion

In this paper, we propose a CNN-based OA-DNN, which aims to recover the high-frequency information of the real-world LR images. Specifically, an orientation feature extraction and channel attention module (OAM) is designed, incorporating three directional convolutional layers (5×1 horizontal conv., 1×5 vertical conv., and 3×3 diagonal conv.), to fully exploit image features extracted in different directions. The directional features are concatenated for learning the complicated nonlinear LR-to-HR mapping. To further enhance the utilization of extracted orientation-aware features, a channel attention mechanism is employed to adaptively compute the channel-wise weights and assign the weights to corresponding channels. Experimental results indicate that the enhanced features can better reconstruct the high-fidelity details. Then, to accelerate the training/testing speed and alleviate memory burden, we reshape the input image via an inverse pixel-shuffle operation (de-pixel-shuffle) into smaller spatial size but deeper depth without losing any information. Extensive experiments demonstrate the priority of our OA-DNN.

In the future, we plan to test our OA-DNN on other benchmarks (e.g., commonly used datasets in SISR - Set5 [3], Set14 [42], B100 [26], Urban100 [13]) to further validate the effectiveness of our method.

References

[1] Image Super-Resolution as Sparse Representation of Raw Image Patches. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008.

[2] Jan Allebach and Ping Wah Wong. Edge-directed interpolation. In *Proceedings of 3rd IEEE International Conference on Image Processing*, volume 3, pages 707–710. IEEE, 1996.

[3] Marco Bevilacqua, Aline Roumy, Christine Guillemot, and Alberi Morel. Low-Complexity Single-Image Super-Resolution based on Nonnegative Neighbor Embedding. In *British Machine Vision Conference (BMVC)*, pages 1–10, 2012.

[4] Yanpeng Cao, Zewei He, Zhangyu Ye, Xin Li, Yanlong Cao, and Jiangxin Yang. Fast and Accurate Single Image Super-Resolution via An Energy-Aware Improved Deep Residual Network. *Signal Processing*, page In Press, 2019.

[5] Xi Cheng, Xiang Li, Jian Yang, and Ying Tai. Sesr: single image super resolution with recursive squeeze and excitation networks. In *2018 24th International Conference on Pattern Recognition (ICPR)*, pages 147–152. IEEE, 2018.

[6] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Learning a deep convolutional network for image super-resolution. In *European conference on computer vision*, pages 184–199. Springer, 2014.

[7] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image Super-Resolution Using Deep Convolutional Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(2):295–307, 2015.

[8] Chao Dong, Chen Change Loy, and Xiaoou Tang. Accelerating the super-resolution convolutional neural network. In *European conference on computer vision*, pages 391–407. Springer, 2016.

[9] Claude E Duchon. Lanczos filtering in one and two dimensions. *Journal of applied meteorology*, 18(8):1016–1022, 1979.

[10] Muhammad Haris, Gregory Shakhnarovich, and Norimichi Ukita. Deep back-projection networks for super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1664–1673, 2018.

[11] Zewei He, Siliang Tang, Jiangxin Yang, Yanlong Cao, Michael Ying Yang, and Yanpeng Cao. Cascaded Deep Networks with Multiple Receptive Fields for Infrared Image Super-Resolution. *IEEE Transactions on Circuits and Systems for Video Technology (Early Access)*, page In Press, 2018.

[12] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018.

[13] Jia-Bin Huang, Abhishek Singh, and Narendra Ahuja. Single image super-resolution from transformed self-exemplars. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5197–5206, 2015.

[14] Zheng Hui, Xiumei Wang, and Xinbo Gao. Two-stage convolutional network for image super-resolution. In *2018 24th International Conference on Pattern Recognition (ICPR)*, pages 2670–2675. IEEE, 2018.

[15] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 675–678. ACM, 2014.

[16] Tao Jiang, Yu Zhang, Xiaojun Wu, Yuan Rao, and Mingquan Zhou. Single Image Super-Resolution via Squeeze and Excitation Network. In *BMVC*, page Accepted, 2018.

[17] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Accurate image super-resolution using very deep convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1646–1654, 2016.

[18] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Deeply-recursive convolutional network for image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1637–1645, 2016.

[19] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[20] Wei-Sheng Lai, Jia-Bin Huang, Narendra Ahuja, and Ming-Hsuan Yang. Deep laplacian pyramid networks for fast and accurate super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 624–632, 2017.

[21] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4681–4690, 2017.

[22] Juncheng Li, Faming Fang, Kangfu Mei, and Guixu Zhang. Multi-scale residual network for image super-resolution. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 517–532, 2018.

[23] Xin Li and Michael T Orchard. New edge-directed interpolation. *IEEE transactions on image processing*, 10(10):1521–1527, 2001.

[24] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 136–144, 2017.

[25] Yue Lu, Yun Zhou, Zhuqing Jiang, Xiaoqiang Guo, and Zixuan Yang. Channel attention and multi-level features fusion for single image super-resolution. *arXiv preprint arXiv:1810.06935*, 2018.

[26] David Martin, Charless Fowlkes, Doron Tal, Jitendra Malik, et al. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *ICCV*. IEEE, 2001.

[27] Joseph Redmon and Ali Farhadi. Yolo9000: Better, faster, stronger. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6517–6525, 2017.

[28] Mehdi SM Sajjadi, Bernhard Scholkopf, and Michael Hirsch. Enhancenet: Single image super-resolution through

automated texture synthesis. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4491–4500, 2017.

[29] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1874–1883, 2016.

[30] Jian Sun, Zongben Xu, and Heung-Yeung Shum. Image super-resolution using gradient profile prior. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008.

[31] Jian Sun, Zongben Xu, and Heung-Yeung Shum. Gradient profile prior and its applications in image super-resolution and enhancement. *IEEE Transactions on Image Processing*, 20(6):1529–1542, 2011.

[32] Ying Tai, Jian Yang, and Xiaoming Liu. Image super-resolution via deep recursive residual network. In *Proceedings of the IEEE Conference on Computer vision and Pattern Recognition*, pages 3147–3155, 2017.

[33] Ying Tai, Jian Yang, Xiaoming Liu, and Chunyan Xu. Memnet: A persistent memory network for image restoration. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4539–4547, 2017.

[34] Radu Timofte, Vincent De Smet, and Luc Van Gool. Anchored neighborhood regression for fast example-based super-resolution. In *Proceedings of the IEEE international conference on computer vision*, pages 1920–1927, 2013.

[35] Radu Timofte, Rasmus Rothe, and Luc Van Gool. Seven ways to improve example-based single image super resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1865–1873, 2016.

[36] Tong Tong, Gen Li, Xiejie Liu, and Qinquan Gao. Image super-resolution using dense skip connections. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4799–4807, 2017.

[37] Thang Vu, Chang D. Yoo, Trung X. Pham, Tung M. Luu, and Cao V. Nguyen. Fast and Efficient Image Quality Enhancement via Desubpixel Convolutional Neural Networks. In *ECCV Workshop*, pages 243–259, 2019.

[38] Yucheng Wang, Jialiang Shen, and Jian Zhang. Deep bi-dense networks for image super-resolution. In *2018 Digital Image Computing: Techniques and Applications (DICTA)*, pages 1–8. IEEE, 2018.

[39] Zhou Wang, Alan Conrad Bovik, Hamid Rahim Sheikh, and Eero P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, apr 2004.

[40] Jianchao Yang, John Wright, Thomas S Huang, and Yi Ma. Image super-resolution via sparse representation. *IEEE transactions on image processing*, 19(11):2861–2873, 2010.

[41] Jiahui Yu, Yuchen Fan, Jianchao Yang, Ning Xu, Zhaowen Wang, Xinchao Wang, and Thomas Huang. Wide activation for efficient and accurate image super-resolution. *arXiv preprint arXiv:1808.08718*, 2018.

[42] Roman Zeyde, Michael Elad, and Matan Protter. On single image scale-up using sparse-representations. In *International conference on curves and surfaces*, pages 711–730. Springer, 2010.

[43] Yulun Zhang, Kunpeng Li, Kai Li, Lichen Wang, Bineng Zhong, and Yun Fu. Image super-resolution using very deep residual channel attention networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 286–301, 2018.

[44] Yulun Zhang, Yapeng Tian, Yu Kong, Bineng Zhong, and Yun Fu. Residual dense network for image super-resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2472–2481, 2018.